

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	6285	version\$4.ti.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/18 22:25
L2	15214	(current\$4 or recent\$4 or new) adj version\$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/18 22:26
L3	760	1 and 2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/18 22:26
L4	180	((current\$4 or recent\$4 or new) adj version\$4) with (application adj program)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/18 22:27
L5	18	1 and 4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/18 22:40
L6	58381	(electronic adj mail adj message) or (e-mail)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/18 22:41
L7	0	5 and 6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/18 22:41
L8	373	(first adj version) near5 data	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/18 22:42
L9	0	5 and 8	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/18 22:42


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☒ The Guide



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used application program versions

Found 92,203 of 161,645

Sort results by


[Save results to a Binder](#)
[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Display results


[Search Tips](#)
☐ Open results in a new window

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

 Relevance scale ☐ ☐ ☐ ☐ ☐

# 1 [The integration of application and system based metrics in a parallel program performance tool](#)



Jeffrey K. Hollingsworth, R. Bruce Irvin, Barton P. Miller

 April 1991 **ACM SIGPLAN Notices , Proceedings of the third ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 26 Issue 7

 Full text available: [pdf\(1.21 MB\)](#)

 Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

# 2 [Integrating noninterfering versions of programs](#)



Susan Horwitz, Jan Prins, Thomas Reps

 July 1989 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 11 Issue 3

 Full text available: [pdf\(3.18 MB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The need to integrate several versions of a program into a common one arises frequently, but it is a tedious and time consuming task to integrate programs by hand. To date, the only available tools for assisting with program integration are variants of text-based differential file comparators; these are of limited utility because one has no guarantees about how the program that is the product of an integration behaves compared to the programs that were integrated.

# 3 [A comparison of application sharing mechanisms in real-time desktop conferencing systems](#)



S. R. Ahuja, J. R. Ensor, S. E. Lucco

 March 1990 **ACM SIGOIS Bulletin , Proceedings of the conference on Office information systems**, Volume 11 Issue 2-3

 Full text available: [pdf\(2.65 MB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Desktop conferencing is a term used to describe real-time, computer-based conferences in which users may share data through their personal computers. In these conferences, the participants may access user-level programs, called application programs, which produce common displays (screens or windows) on their computers. Because each participant may give input to the application program and sees its resulting output as though the program were executing on his or her local computer, these appl ...

# 4


[Applicative style programming, program transformation, and list operators](#)

Philip Wadler

October 1981 **Proceedings of the 1981 conference on Functional programming languages and computer architecture**Full text available:  pdf(698.23 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

An important feature of the applicative style is the use of operators that package common patterns of computation. For example, the list operator map applies a function to every element of a list. Practical use of this style has been hampered by the fact that it can be very inefficient to execute. One remedy for this situation is to use source-to-source program transformation to convert applicative style programs to more efficient equivalents. This paper ex ...

5 Continuous program optimization: A case study

Thomas Kistler, Michael Franz

July 2003 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 25 Issue 4Full text available:  pdf(877.67 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [review](#)

Much of the software in everyday operation is not making optimal use of the hardware on which it actually runs. Among the reasons for this discrepancy are hardware/software mismatches, modularization overheads introduced by software engineering considerations, and the inability of systems to adapt to users' behaviors. A solution to these problems is to delay code generation until load time. This is the earliest point at which a piece of software can be fine-tuned to the actual capabilities of the ...

**Keywords:** Dynamic code generation, continuous program optimization, dynamic reoptimization

6 Technical papers: Learning programs from traces using version space algebra

Tessa Lau, Pedro Domingos, Daniel S. Weld

October 2003 **Proceedings of the international conference on Knowledge capture**Full text available:  pdf(338.14 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

While existing learning techniques can be viewed as inducing programs from examples, most research has focused on rather narrow classes of programs, e.g., decision trees or logic rules. In contrast, most of today's programs are written in languages such as C++ or Java. Thus, many tasks we wish to automate (e.g. programming by demonstration and software reverse engineering) might be best formulated as induction of code in a procedural language. In this paper we apply version space algebra [10] to ...

7 Application-level checkpointing for shared memory programs

Greg Bronevetsky, Daniel Marques, Keshav Pingali, Peter Szwed, Martin Schulz

October 2004 **Proceedings of the 11th international conference on Architectural support for programming languages and operating systems**, Volume 32 , 38 , 39 Issue 5 , 5 , 11Full text available:  pdf(235.77 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Trends in high-performance computing are making it necessary for long-running applications to tolerate hardware faults. The most commonly used approach is checkpoint and restart (CPR) - the state of the computation is saved periodically on disk, and when a failure occurs, the computation is restarted from the last saved state. At present, it is the responsibility of the programmer to instrument applications for CPR. Our group is investigating the use of compiler technology to instrument codes to ...

**Keywords:** checkpointing, fault-tolerance, openMP, shared-memory programs

8 Contention elimination by replication of sequential sections in distributed shared memory programs



Honghui Lu, Alan L. Cox, Willy Zwaenepoel

June 2001 **ACM SIGPLAN Notices , Proceedings of the eighth ACM SIGPLAN symposium on Principles and practices of parallel programming**, Volume 36 Issue 7

Full text available: pdf(173.49 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In shared memory programs contention often occurs at the transition between a sequential and a parallel section of the code. As all threads start executing the parallel section, they often access data just modified by the thread that executed the sequential section, causing a flurry of data requests to converge on that processor. We address this problem in a software distributed shared memory system by replicating the execution of the sequential sections on all pr ...

9 A framework for remote dynamic program optimization



Michael J. Voss, Rudolf Eigenmann

January 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN workshop on Dynamic and adaptive compilation and optimization**, Volume 35 Issue 7

Full text available: pdf(1.12 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Dynamic program optimization allows programs to be generated that are highly tuned for a given environment and input data set. Optimization techniques can be applied and re-applied as program and machine characteristics are discovered and change. In most dynamic optimization and compilation frameworks, the time spent in code generation and optimization must be minimized since it is directly reflected in the total program execution time. We propose a generic framework for remote dynamic progra ...

10 Measuring the dynamic behaviour of AspectJ programs



Bruno Dufour, Christopher Goard, Laurie Hendren, Oege de Moor, Ganesh Sittampalam, Clark Verbrugge

October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN Conference on Object-oriented programming, systems, languages, and applications**, Volume 39 Issue 10

Full text available: pdf(226.86 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper proposes and implements a rigorous method for studying the dynamic behaviour of AspectJ programs. As part of this methodology several new metrics specific to AspectJ programs are proposed and tools for collecting the relevant metrics are presented. The major tools consist of: (1) a modified version of the AspectJ compiler that tags bytecode instructions with an indication of the cause of their generation, such as a particular feature of AspectJ; and (2) a modified version of the \*J ...

**Keywords:** AspectJ, aspect-oriented programming, dynamic metrics, java, optimization, performance, program analysis

11 SPLASH: Stanford parallel applications for shared-memory



Jaswinder Pal Singh, Wolf-Dietrich Weber, Anoop Gupta

March 1992 **ACM SIGARCH Computer Architecture News**, Volume 20 Issue 1

Full text available: pdf(3.04 MB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

We present the Stanford Parallel Applications for Shared-Memory (SPLASH), a set of parallel

applications for use in the design and evaluation of shared-memory multiprocessing systems. Our goal is to provide a suite of realistic applications that will serve as a well-documented and consistent basis for evaluation studies. We describe the applications currently in the suite in detail, discuss some of their important characteristics, and explore their behavior by running them on a real multiprocess ...

12 Eliminating synchronization overhead in automatically parallelized programs using dynamic feedback



Pedro C. Diniz, Martin C. Rinard

May 1999 **ACM Transactions on Computer Systems (TOCS)**, Volume 17 Issue 2

Full text available: pdf(244.57 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This article presents dynamic feedback, a technique that enables computations to adapt dynamically to different execution environments. A compiler that uses dynamic feedback produces several different versions of the same source code; each version uses a different optimization policy. The generated code alternately performs sampling phases and production phases. Each sampling phase measures the overhead of each version in the current environment. Each production phase uses the version with ...

**Keywords:** parallel computing, parallelizing compilers

13 Application and architectural bottlenecks in large scale distributed shared memory machines



Chris Holt, Jaswinder Pal Singh, John Hennessy

May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture**, Volume 24 Issue 2

Full text available: pdf(1.55 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Many of the programming challenges encountered in small to moderate-scale hardware cache-coherent shared memory machines have been extensively studied. While work remains to be done, the basic techniques needed to efficiently program such machines have been well explored. Recently, a number of researchers have presented architectural techniques for scaling a cache coherent shared address space to much larger processor counts. In this paper, we examine the extent to which applications can achieve ...

14 Transformation of data traversals and operations in application programs to account for semantic changes of databases



Stanley Y. W. Su, Herman Lam, Der Her Lo


June 1981 **ACM Transactions on Database Systems (TODS)**, Volume 6 Issue 2

Full text available: pdf(3.00 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper addresses the problem of application program conversion to account for changes in database semantics that result in changes in the schema and database contents. With the observation that the existing data models can be viewed as alternative ways of modeling the same database semantics, a methodology of application program analysis and conversion based on an existing-DBMS-model-and schema-independent representation of both the database and programs is presented. In this methodolog ...


**Keywords:** access pattern, application program conversion, database changes, semantic data model, transformation rules

- 15 On using SCALEA for performance analysis of distributed and parallel programs   
Hong-Linh Truong, Thomas Fahringer, Georg Madsen, Allen D. Malony, Hans Moritsch, Sameer Shende  
November 2001 **Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM)**


Full text available:  pdf(229.70 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In this paper we give an overview of SCALEA, which is a new performance analysis tool for OpenMP, MPI, HPF, and mixed parallel/distributed programs. SCALEA instruments, executes and measures programs and computes a variety of performance overheads based on a novel overhead classification. Source code and HW-profiling is combined in a single system which significantly extends the scope of possible overheads that can be measured and examined, ranging from HW-counters, such as the number of cache m ...

**Keywords:** distributed and parallel systems, performance analysis, performance overhead classification

- 16 Developing MFC application NAVAJO Word Processor Version 2.0   
Scott Hall, Stephen James, Alesha Platero, Jim Reihsen, Gary Ross, Kenny Smith, Matt Yazzie, Stephanie Wimbish, Tim Reeves  
October 2000 **Journal of Computing Sciences in Colleges**, Volume 16 Issue 1


Full text available:  pdf(84.36 KB) Additional Information: [full citation](#), [references](#), [index terms](#)

- 17 Effective fine-grain synchronization for automatically parallelized programs using optimistic synchronization primitives   
Martin C. Rinard  
November 1999 **ACM Transactions on Computer Systems (TOCS)**, Volume 17 Issue 4

Full text available:  pdf(637.69 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This article presents our experience using optimistic synchronization to implement fine-grain atomic operations in the context of a parallelizing compiler for irregular, object-based computations. Our experience shows that the synchronization requirements of these programs differ significantly from those of traditional parallel computations, which use loop nests to access dense matrices using affine access functions. In addition to coarse-grain barrier synchronization, our irregular comput ...

**Keywords:** atomic operations commutativity analysis, optimistic synchronization, parallel computing, parallelizing compilers, synchronization

- 18 Under CoVer: the implementation of a contextual version server for hypertext applications   
Anja Haake  
September 1994 **Proceedings of the 1994 ACM European conference on Hypermedia technology**

Full text available:  pdf(1.61 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

At GMD-IPSI we are developing CoVer, a contextual version server for hypertext applications. Another characterization of CoVer is that CoVer integrates state-oriented versioning concepts with task-oriented versioning concepts. While other version models in general support only one of these groups of concepts, we argue that the explicit

composition of versions of complex hypertext networks has to be complemented by automatic version creation in the context of tasks or jobs p ...

**Keywords:** alternatives, implementation techniques, publishing applications, state-oriented versioning, task-oriented versioning, versioning

**19** A self-applicable partial evaluator for the lambda calculus: correctness and pragmatics 

Carsten K. Gomard

April 1992 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,

Volume 14 Issue 2

Full text available:  [pdf\(1.72 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We describe theoretical and a few practical aspects of an implemented self-applicable partial evaluator for the untyped lambda calculus with constants, conditionals, and a fixed point operator. The purpose of this paper is first to announce the existence of (and to describe) a partial evaluator that is both higher-order and self-applicable; second to describe a surprisingly simple solution to the central problem of binding time analysis, and third to prove that the partial evalua ...

**Keywords:** compiler generation, lambda calculus, partial evaluation, self-application

**20** Conversion technology, an assessment 

James P. Fry

July 1981 **ACM SIGMIS Database , ACM SIGMOD Record**, Volume 12,13 , 12 Issue 4,1 , 2

Full text available:  [pdf\(2.36 MB\)](#) Additional Information: [full citation](#), [references](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☒ The Guide



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used application program different versions

Found 115,726 of 161,645

Sort results by


[Save results to a Binder](#)
[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Display results


[Search Tips](#)
☐ Open results in a new window

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

 Relevance scale ☐ ☐ ☐ ☐ ☐

### 1 [Continuous program optimization: A case study](#)

Thomas Kistler, Michael Franz

 July 2003 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,  
Volume 25 Issue 4

Full text available: pdf(877.67 KB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [review](#)

Much of the software in everyday operation is not making optimal use of the hardware on which it actually runs. Among the reasons for this discrepancy are hardware/software mismatches, modularization overheads introduced by software engineering considerations, and the inability of systems to adapt to users' behaviors. A solution to these problems is to delay code generation until load time. This is the earliest point at which a piece of software can be fine-tuned to the actual capabilities of the ...

**Keywords:** Dynamic code generation, continuous program optimization, dynamic reoptimization

### 2 [Integrating noninterfering versions of programs](#)

Susan Horwitz, Jan Prins, Thomas Reps

 July 1989 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,  
Volume 11 Issue 3

Full text available: pdf(3.18 MB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The need to integrate several versions of a program into a common one arises frequently, but it is a tedious and time consuming task to integrate programs by hand. To date, the only available tools for assisting with program integration are variants of text-based differential file comparators; these are of limited utility because one has no guarantees about how the program that is the product of an integration behaves compared to the programs that were integrated.

### 3 [A comparison of application sharing mechanisms in real-time desktop conferencing systems](#)

S. R. Ahuja, J. R. Ensor, S. E. Lucco

 March 1990 **ACM SIGOIS Bulletin , Proceedings of the conference on Office information systems**, Volume 11 Issue 2-3

Full text available: pdf(2.65 MB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Desktop conferencing is a term used to describe real-time, computer-based conferences in which users may share data through their personal computers. In these conferences, the participants may access user-level programs, called application programs, which produce common displays (screens or windows) on their computers. Because each participant may give input to the application program and sees its resulting output as though the program were executing on his or her local computer, these appl ...

#### 4 Program Transformation Systems

H. Partsch, R. Steinbrüggen

September 1983 **ACM Computing Surveys (CSUR)**, Volume 15 Issue 3

Full text available:  pdf(3.00 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)



#### 5 Techniques for reducing consistency-related communication in distributed shared-memory systems

John B. Carter, John K. Bennett, Willy Zwaenepoel

August 1995 **ACM Transactions on Computer Systems (TOCS)**, Volume 13 Issue 3

Full text available:  pdf(2.86 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)



Distributed shared memory (DSM) is an abstraction of shared memory on a distributed-memory machine. Hardware DSM systems support this abstraction at the architecture level; software DSM systems support the abstraction within the runtime system. One of the key problems in building an efficient software DSM system is to reduce the amount of communication needed to keep the distributed memories consistent. In this article we present four techniques for doing so: software release consistency; m ...

**Keywords:** cache consistency protocols, distributed shared memory, memory models, release consistency, virtual shared memory

#### 6 A hierarchical structure for fault tolerant reactive programs

Andrea Clematis, Vittoria Gianuzzi

March 1993 **Proceedings of the 1993 ACM/SIGAPP symposium on Applied computing: states of the art and practice**

Full text available:  pdf(613.91 KB)

Additional Information: [full citation](#), [references](#), [index terms](#)



**Keywords:** backward error recovery, concurrent programming, software fault tolerance, transaction based systems

#### 7 Eliminating synchronization overhead in automatically parallelized programs using dynamic feedback

Pedro C. Diniz, Martin C. Rinard

May 1999 **ACM Transactions on Computer Systems (TOCS)**, Volume 17 Issue 2

Full text available:  pdf(244.57 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)



This article presents dynamic feedback, a technique that enables computations to adapt dynamically to different execution environments. A compiler that uses dynamic feedback produces several different versions of the same source code; each version uses a different optimization policy. The generated code alternately performs sampling phases and production phases. Each sampling phase measures the overhead of each version in the

current environment. Each production phase uses the version with ...

**Keywords:** parallel computing, parallelizing compilers

8 Tracing application program execution on the Cray X-MP and Cray 2

Allen D. Malony, John L. Larson, Daniel A. Reed

November 1990 **Proceedings of the 1990 ACM/IEEE conference on Supercomputing**

Full text available:  [pdf\(1.28 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#)

Important insights into program operation can be gained by observing dynamic execution behavior. Unfortunately, many high-performance machines provide execution profile summaries as the only tool for performance investigation. We have developed a tracing library for the Cray X-MP and Cray 2 supercomputers that supports the low-overhead capture of execution events for sequential and multitasked programs. This library has been extended to use the automatic instrumentation facilities on these machi ...

9 Efficiently serving dynamic data at highly accessed web sites

James R. Challenger, Paul Dantzig, Arun Iyengar, Mark S. Squillante, Li Zhang

April 2004 **IEEE/ACM Transactions on Networking (TON)**, Volume 12 Issue 2

Full text available:  [pdf\(499.05 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We present architectures and algorithms for efficiently serving dynamic data at highly accessed Web sites together with the results of an analysis motivating our design and quantifying its performance benefits. This includes algorithms for keeping cached data consistent so that dynamic pages can be cached at the Web server and dynamic content can be served at the performance level of static content. We show that our system design is able to achieve cache hit ratios close to 100% for cached data ...

**Keywords:** caching, dynamic content, performance analysis, prefetching, stochastic models, web sites

10 Using Applications of Data Versioning in Database Application Development

Ramkrishna Chatterjee, Gopalan Arun, Sanjay Agarwal, Ben Speckhard, Ramesh Vasudevan

May 2004 **Proceedings of the 26th International Conference on Software Engineering**

Full text available:  [pdf\(166.57 KB\)](#)

Additional Information: [full citation](#), [abstract](#)

Database applications such as enterprise resource planning systems and customer relationship management systems are widely used software systems. Development and testing of database applications is difficult because the program execution depends on the persistent state stored in the database. In this paper we show that how versioning of the persistent data stored in the database can solve some critical problems in the development and testing of database applications can be solved by vers ...

11 Library support for hierarchical multi-processor tasks

Thomas Rauber, Gudula Rünger

November 2002 **Proceedings of the 2002 ACM/IEEE conference on Supercomputing**

Full text available:  [pdf\(132.76 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)

The paper considers the modular programming with hierarchically structured multi-processor tasks on top of SPMD tasks for distributed memory machines. The parallel execution requires a corresponding decomposition of the set of processors into a hierarchical group structure onto which the tasks are mapped. This results in a multi-level group SPMD computation model with varying processor group structures. The advantage of

this kind of mixed task and data parallelism is a potential to reduce the co ...

**Keywords:** distributed memory, hierarchical decomposition of processor sets, library support, mixed task and data parallelism, multilevel group SPMD, multiprocessor tasks

12 Effective fine-grain synchronization for automatically parallelized programs using optimistic synchronization primitives

Martin C. Rinard

November 1999 **ACM Transactions on Computer Systems (TOCS)**, Volume 17 Issue 4

Full text available:  pdf(637.69 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This article presents our experience using optimistic synchronization to implement fine-grain atomic operations in the context of a parallelizing compiler for irregular, object-based computations. Our experience shows that the synchronization requirements of these programs differ significantly from those of traditional parallel computations, which use loop nests to access dense matrices using affine access functions. In addition to coarse-grain barrier synchronization, our irregular comput ...

**Keywords:** atomic operations commutativity analysis, optimistic synchronization, parallel computing, parallelizing compilers, synchronization

13 Workshop on Dynamic Analysis (WODA): Selective capture and replay of program executions

Alessandro Orso, Bryan Kennedy

May 2005 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2005 workshop on Dynamic analysis WODA '05**, Volume 30 Issue 4

Full text available:  pdf(167.30 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

In this paper, we present a technique for selective capture and replay of program executions. Given an application, the technique allows for (1) selecting a subsystem of interest, (2) capturing at runtime all the interactions between such subsystem and the rest of the application, and (3) replaying the recorded interactions on the subsystem in isolation. The technique can be used in several scenarios. For example, it can be used to generate test cases from users' executions, by capturing and col ...

14 Combination of an optimization model for hardware selection with data determination methods

A.-W. Scheer

July 1977 **ACM SIGMETRICS Performance Evaluation Review**, Volume 6 Issue 3

Full text available:  pdf(715.80 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

The selection of an EDP configuration often fixes a firm to a single manufacturer for a long time and the capabilities of the chosen computer will continually influence the firm's organization. Only few approaches exist to give assistance to the investors by developing useful decision models based on the investment theory /11, 12/. The hardware selection methods /4, 13/ used up to now, like benchmark tests, don't meet these demands. In this paper an investment model based on mathematical program ...

15 Application and architectural bottlenecks in large scale distributed shared memory machines

Chris Holt, Jaswinder Pal Singh, John Hennessy

May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture**, Volume 24 Issue 2

Full text available:  pdf(1.55 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Many of the programming challenges encountered in small to moderate-scale hardware cache-coherent shared memory machines have been extensively studied. While work remains to be done, the basic techniques needed to efficiently program such machines have been well explored. Recently, a number of researchers have presented architectural techniques for scaling a cache coherent shared address space to much larger processor counts. In this paper, we examine the extent to which applications can achieve ...

#### 16 [Versioning and configuration management in an object-oriented data model](#)

Edward Sciore

January 1994 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 3 Issue 1

Full text available:  pdf(1.57 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)


Many database applications require the storage and manipulation of different versions of data objects. To satisfy the diverse needs of these applications, current database systems support versioning at a very low level. This article demonstrates that application-independent versioning can be supported at a significantly higher level. In particular, we extend the EXTRA data model and EXCESS query language so that configurations can be specified conceptually and non-procedurally. We also show how ...

**Keywords:** EXTRA/EXCESS data models, generic and specific references, query language, semantically based configuration specifications

#### 17 [Effective fine-grain synchronization for automatically parallelized programs using optimistic synchronization primitives](#)

Martin Rinard

June 1997 **ACM SIGPLAN Notices , Proceedings of the sixth ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 32 Issue 7

Full text available:  pdf(1.38 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

As shared-memory multiprocessors become the dominant commodity source of computation, parallelizing compilers must support mainstream computations that manipulate irregular, pointer-based data structures such as lists, trees and graphs. Our experience with a parallelizing compiler for this class of applications shows that their synchronization requirements differ significantly from those of traditional parallel computations. Instead of coarse-grain barrier synchronization, irregular computations ...

#### 18 [Relative debugging: a new methodology for debugging scientific applications](#)

David Abramson, Ian Foster, John Michalakes, Rok Sosič

November 1996 **Communications of the ACM**, Volume 39 Issue 11

Full text available:  pdf(462.99 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

#### 19 [Implementation and Evaluation of a Scalable Application-Level Checkpoint-Recovery Scheme for MPI Programs](#)

Martin Schulz, Greg Bronevetsky, Rohit Fernandes, Daniel Marques, Keshav Pingali, Paul Stodghill

November 2004 **Proceedings of the 2004 ACM/IEEE conference on Supercomputing**

Full text available:  pdf(183.27 KB) Additional Information: [full citation](#), [abstract](#)

The running times of many computational science applications are much longer than the

mean-time-to-failure of current high-performance computing platforms. To run to completion, such applications must tolerate hardware failures. Checkpoint-and-restart (CPR) is the most commonly used scheme for accomplishing this - the state of the computation is saved periodically on stable storage, and when a hardware failure is detected, the computation is restarted from the most recently saved state. Most aut ...

**20** GNATDIST: a configuration language for distributed Ada 95 applications

Yvon Kermarrec, Laurent Nana, Laurent Pautet

December 1996 **Proceedings of the conference on TRI-Ada '96: disciplined software development with Ada**

Full text available:  pdf(999.63 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

**Keywords:** GNAT, configuration language, distributed systems programming

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.  
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)



Welcome United States Patent and Trademark Office

Search Results

[BROWSE](#)[SEARCH](#)[IEEE XPLORE GUIDE](#)

Results for "((application&lt;sentence&gt;program&lt;sentence&gt;versions)&lt;in&gt;metadata)"

e-mail

Your search matched **55** of **1235066** documents.A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

## » Search Options

[View Session History](#)[New Search](#)

## Modify Search

((application&lt;sentence&gt;program&lt;sentence&gt;versions)&lt;in&gt;metadata)

☐ Check to search only within this results setDisplay Format: ☒ Citation ☐ Citation & Abstract

## » Key

IEEE JNL IEEE Journal or Magazine

IEEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

Select Article Information

View: 1-

**1. Experience with pipelined multiple instruction streams**

Jordan, H.F.;  
 Proceedings of the IEEE  
 Volume 72, Issue 1, Jan. 1984 Page(s):113 - 123  
[AbstractPlus](#) | Full Text: [PDF](#)(1076 KB) IEEE JNL

**2. IEEE's Posix: making progress**

Kuhn, D.R.;  
 Spectrum, IEEE  
 Volume 28, Issue 12, Dec. 1991 Page(s):36 - 39  
 Digital Object Identifier 10.1109/6.106103  
[AbstractPlus](#) | Full Text: [PDF](#)(536 KB) IEEE JNL

**3. Evaluation and comparison of fault-tolerant software techniques**

Hudak, J.J.; Suh, B.-H.; Siewiorek, D.P.; Segall, Z.;  
 Reliability, IEEE Transactions on  
 Volume 42, Issue 2, June 1993 Page(s):190 - 204  
 Digital Object Identifier 10.1109/24.229487  
[AbstractPlus](#) | Full Text: [PDF](#)(1372 KB) IEEE JNL

**4. Improving the N-version programming process through the evolution of a paradigm**

Lyu, M.R.; He, Y.-T.;  
 Reliability, IEEE Transactions on  
 Volume 42, Issue 2, June 1993 Page(s):179 - 189  
 Digital Object Identifier 10.1109/24.229486  
[AbstractPlus](#) | Full Text: [PDF](#)(864 KB) IEEE JNL

**5. System reliability analysis of an N-version programming application**

Dugan, J.B.; Lyu, M.R.;  
 Reliability, IEEE Transactions on  
 Volume 43, Issue 4, Dec. 1994 Page(s):513 - 519  
 Digital Object Identifier 10.1109/24.370232  
[AbstractPlus](#) | Full Text: [PDF](#)(528 KB) IEEE JNL

**6. A coverage analysis tool for the effectiveness of software testing**

Lyu, M.R.; Horgan, J.R.; London, S.;

Reliability, IEEE Transactions on  
Volume 43, Issue 4, Dec. 1994 Page(s):527 - 535  
Digital Object Identifier 10.1109/24.370230  
[AbstractPlus](#) | Full Text: [PDF\(772 KB\)](#) IEEE JNL



**7. Trends in reliability and test strategies**

Varadan, G.S.;  
Software, IEEE  
Volume 12, Issue 3, May 1995 Page(s):10  
Digital Object Identifier 10.1109/52.382175  
[AbstractPlus](#) | Full Text: [PDF\(88 KB\)](#) IEEE JNL



**8. A national-scale authentication infrastructure**

Butler, R.; Welch, V.; Engert, D.; Foster, I.; Tuecke, S.; Volmer, J.; Kesselman, Computer  
Volume 33, Issue 12, Dec. 2000 Page(s):60 - 66  
Digital Object Identifier 10.1109/2.889094  
[AbstractPlus](#) | [References](#) | Full Text: [PDF\(144 KB\)](#) IEEE JNL



**9. Samurai Romanesque, J2ME, and the battle for mobile cyberspace**

Krikke, J.;  
Computer Graphics and Applications, IEEE  
Volume 23, Issue 1, Jan.-Feb. 2003 Page(s):16 - 23  
Digital Object Identifier 10.1109/MCG.2003.1159608  
[AbstractPlus](#) | Full Text: [PDF\(2025 KB\)](#) IEEE JNL



**10. Faulty version recovery in object-oriented N-version programming**

Romanovsky, A.;  
Software, IEEE Proceedings- [see also Software Engineering, IEEE Proceedings]  
Volume 147, Issue 3, June 2000 Page(s):81 - 90  
Digital Object Identifier 10.1049/ip-sen:20000679  
[AbstractPlus](#) | Full Text: [PDF\(764 KB\)](#) IEEE JNL



**11. Experimental and theoretical study of electrical machines with bulk HTS**

Kovalev, K.L.; Kovalev, L.K.; Ilushin, K.V.; Larionov, S.A.; Poltavets, V.N.; Gav  
Power Electronics and Motion Control Conference, 2004. IPEMC 2004. The 4th  
Volume 2, 14-16 Aug. 2004 Page(s):568 - 570 Vol.2  
[AbstractPlus](#) | Full Text: [PDF\(575 KB\)](#) IEEE CNF



**12. Design, implementation and performance evaluation of GridRPC program middleware for a large-scale computational grid**









Tanaka, Y.; Takemiya, H.; Nakada, N.; Sekiguchi, S.;  
Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on  
8 Nov. 2004 Page(s):298 - 305  
Digital Object Identifier 10.1109/GRID.2004.20  
[AbstractPlus](#) | Full Text: [PDF\(200 KB\)](#) IEEE CNF



**13. Parallel program graphical design with program execution control based application states**

Tudruj, M.; Borkowski, J.; Kopanski, D.;  
Parallel and Distributed Computing, 2004. Third International Symposium on/A  
and Tools for Parallel Computing on Heterogeneous Networks, 2004. Third Int  
Workshop on  
5-7 July 2004 Page(s):240 - 247  
Digital Object Identifier 10.1109/ISPDC.2004.38  
[AbstractPlus](#) | Full Text: [PDF\(248 KB\)](#) IEEE CNF

**14. SimSnap: fast-forwarding via native execution and application-level chec**

-  Szwed, P.K.; Marques, D.; Buels, R.M.; McKee, S.A.; Schulz, M.;  
Interaction between Compilers and Computer Architectures, 2004. INTERACT  
Workshop on  
15 Feb. 2004 Page(s):65 - 74  
Digital Object Identifier 10.1109/INTERA.2004.1299511  
[AbstractPlus](#) | Full Text: [PDF](#)(1355 KB) [IEEE CNF](#)
-  **15. A fast algorithm for the exhaustive analysis of 12-nucleotide-long DNA sequences**  
Applications to human genomics  
Arnau, V.; Marin, I.;  
Parallel and Distributed Processing Symposium, 2003. Proceedings. International  
22-26 April 2003 Page(s):7 pp.  
Digital Object Identifier 10.1109/IPDPS.2003.1213287  
[AbstractPlus](#) | Full Text: [PDF](#)(300 KB) [IEEE CNF](#)
-  **16. On the viability of component frameworks for high performance distributed case study**  
Kurzyniec, D.; Sunderam, V.; Migliardi, M.;  
High Performance Distributed Computing, 2002. HPDC-11 2002. Proceedings  
International Symposium on  
23-26 July 2002 Page(s):275 - 283  
Digital Object Identifier 10.1109/HPDC.2002.1029927  
[AbstractPlus](#) | Full Text: [PDF](#)(374 KB) [IEEE CNF](#)
-  **17. A software infrastructure for distributed computing based on DCOM**  
Milanovic, N.; Mornar, V.;  
Information Technology Interfaces, 2001. ITI 2001. Proceedings of the 23rd International  
Conference on  
19-22 June 2001 Page(s):63 - 68 vol.1  
Digital Object Identifier 10.1109/ITI.2001.937998  
[AbstractPlus](#) | Full Text: [PDF](#)(440 KB) [IEEE CNF](#)
-  **18. Augmented CPU reservations: towards predictable execution on general operating systems**  
Regehr, J.; Stankovic, J.A.;  
Real-Time Technology and Applications Symposium, 2001. Proceedings. Seventh  
30 May-1 June 2001 Page(s):141 - 148  
Digital Object Identifier 10.1109/RTTAS.2001.929880  
[AbstractPlus](#) | Full Text: [PDF](#)(676 KB) [IEEE CNF](#)
-  **19. Optimal allocation of electronic content**  
Cidon, I.; Kutten, S.; Soffer, R.;  
INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and  
Communications Societies. Proceedings. IEEE  
Volume 3, 22-26 April 2001 Page(s):1773 - 1780 vol.3  
Digital Object Identifier 10.1109/INFCOM.2001.916675  
[AbstractPlus](#) | Full Text: [PDF](#)(836 KB) [IEEE CNF](#)
-  **20. Building dependable software for critical applications: multi-version software good version**  
Townend, P.; Jie Xu; Munro, M.;  
Object-Oriented Real-Time Dependable Systems, 2001. Proceedings. Sixth International  
Workshop on  
8-10 Jan. 2001 Page(s):103 - 110  
Digital Object Identifier 10.1109/WORDS.2001.945120  
[AbstractPlus](#) | Full Text: [PDF](#)(680 KB) [IEEE CNF](#)
-  **21. End-user programming in a structured dialogue environment: the GIPSE**  
Patry, G.; Girard, P.;

Human-Centric Computing Languages and Environments, 2001. Proceedings on

5-7 Sept. 2001 Page(s):212 - 219

Digital Object Identifier 10.1109/HCC.2001.995261

[AbstractPlus](#) | Full Text: [PDF\(526 KB\)](#) IEEE CNF



**22. Performance optimization for data intensive grid applications**

Beynon, M.D.; Sussman, A.; Catalyurek, U.; Kurc, T.; Saltz, J.;

Active Middleware Services, 2001. Third Annual International Workshop on 6 Aug. 2001 Page(s):97 - 105

[AbstractPlus](#) | Full Text: [PDF\(848 KB\)](#) IEEE CNF



**23. Implementation of a WNW within the JTRS operating environment using**

Anderson, J.; Stevens, J.; Mabe, F.;

Military Communications Conference, 2001. MILCOM 2001. Communications Centric Operations: Creating the Information Force. IEEE

Volume 2, 28-31 Oct. 2001 Page(s):972 - 976 vol.2

Digital Object Identifier 10.1109/MILCOM.2001.985984

[AbstractPlus](#) | Full Text: [PDF\(134 KB\)](#) IEEE CNF



**24. Coware pipelining for exploiting Intellectual properties and software code based designs**

Hoon Choi; In-Cheol Park;

ASIC/SOC Conference, 2000. Proceedings. 13th Annual IEEE International 13-16 Sept. 2000 Page(s):153 - 157

Digital Object Identifier 10.1109/ASIC.2000.880693

[AbstractPlus](#) | Full Text: [PDF\(444 KB\)](#) IEEE CNF



**25. Matching software fault tolerance and application needs**

Shokri, E.; Hecht, H.;

High-Assurance Systems Engineering Symposium, 1998. Proceedings. Third 13-14 Nov. 1998 Page(s):248 - 251

Digital Object Identifier 10.1109/HASE.1998.731622

[AbstractPlus](#) | Full Text: [PDF\(36 KB\)](#) IEEE CNF



View: 1-

Indexed by  
 Inspec

[Help](#) [Contact Us](#) [Privacy & ;](#)

© Copyright 2005 IEEE --